SMT4STECTL: Satisfiability-Driven Synthesis from Specifications in Strategic Timed Existential CTL

Demonstration Track

Artur Niewiadomski
University of Siedlce,
Faculty of Exact and Natural Sciences
Siedlce, Poland
artur.niewiadomski@uws.edu.pl

Mateusz Przychodzki
University of Siedlce,
Faculty of Exact and Natural Sciences
Siedlce, Poland
mateusz.przychodzki@uws.edu.pl

Magdalena Kacprzak
Faculty of Computer Science,
Bialystok University of Technology
Bialystok, Poland
m.kacprzak@pb.edu.pl

Wojciech Penczek
Institute of Computer Science,
Polish Academy of Sciences
Warsaw, Poland
wpenczek@gmail.com

Andrzej Zbrzezny
Faculty of Science and Technology,
Jan Dlugosz University
in Czestochowa, Poland
andrzej.zbrzezny@gmail.com

ABSTRACT

We present SMT4STECTL, a tool for automatically synthesising real-time multi-agent systems from strategic, time-dependent STECTL specifications. Using SMT solving and bounded model checking, the tool checks specification consistency and, when satisfiable, constructs a compliant system model. SMT4STECTL enables prototyping of strategic interactions under timing constraints, shown on drone surveillance benchmark, and accessible via a web interface.

KEYWORDS

model synthesis; Strategic Timed Existential CTL; SMT; BMC

ACM Reference Format:
Artur Niewiadomski, Mateusz Przychodzki, Magdalena Kacprzak, Wojciech Penczek, and Andrzej Zbrzezny. 2026. SMT4STECTL: Satisfiability-Driven Synthesis from Specifications in Strategic Timed Existential CTL: Demonstration Track. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/YCFH5927

1 INTRODUCTION

Model synthesis is a key step in real-time system design, where correctness depends on logical behaviour and timing constraints. It aims to construct models satisfying functional specifications under temporal requirements. Our tool supports synthesis and satisfiability for Strategic Timed Computation Tree Logic (STCTL), an extension of TCTL with strategic operators [4, 5]. While model checking for a fragment of STCTL was explored [4, 5], the general satisfiability problem remains undecidable [1, 8, 10, 30]. We therefore focus on the existential fragment of STCTL (called STECTL) and propose a partially terminating algorithm for bounded satisfiability checking, which builds on SMTL [18]. A key contribution is parametric reasoning at both the model and formula levels — introduced here for the first time — reducing verification to a search for satisfying valuations over families of systems. To our knowledge, SMT4STECTL is the first tool supporting parametric bounded synthesis for a strategic timed branching-time logic.

2 APPLICATION DOMAIN

Model synthesis plays a central role in multi-agent and real-time AI systems, where autonomous entities must coordinate decisions under shared constraints. Similar challenges arise in autonomous robots and drones, intelligent transportation, cyber-physical and embedded systems, and multi-agent scheduling, where safe coordination and conflict-free resource allocation are essential [7, 13, 15, 19, 28]. In these domains, the key question is the existence of a feasible real-time strategy rather than the correctness of all behaviours. STECTL-based model synthesis directly addresses this need, enabling automated design and validation of systems.

3 TOOL COMPARISON

Reasoning about strategies in multi-agent systems has led to verification tools, mostly focused on model checking. MCMAS [24] and MCMAS-SLK [11, 12] support strategic reasoning but not synthesis or real-time constraints, while MOCHA [2, 35] and STV [20, 22] provide strategic verification, including imperfect information, without model synthesis. Recent SMT-based approaches MsATL [16, 26], SGSAT [17], and SMT4SMTL [18, 27] support satisfiability and synthesis for a strategic timed logic based on LTL [31]. Some general-purpose tools have been extended: IMITATOR [3, 29] supports fragments of STCTL via encoding, limited to flat formulae and fixed system structures, while Maude [6] supports full STCTL model checking and timing synthesis but only limited strategy and model synthesis. STV+FLY [21] enables scalable on-the-fly strategy synthesis under imperfect information but is restricted to untimed settings. Our tool advances the state of the art by supporting bounded satisfiability, strategy-aware synthesis, and full model construction for strategic timed logic, handling nested formulae and parametric constraints via SMT-based bounded model checking.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/YCFH5927

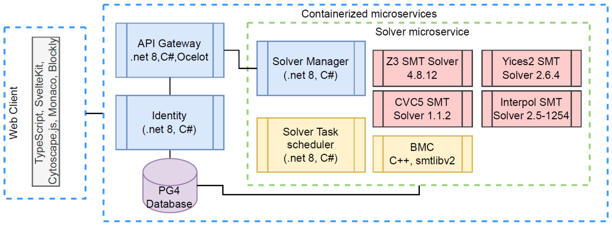

Figure 1: SMT4STECTL architecture.

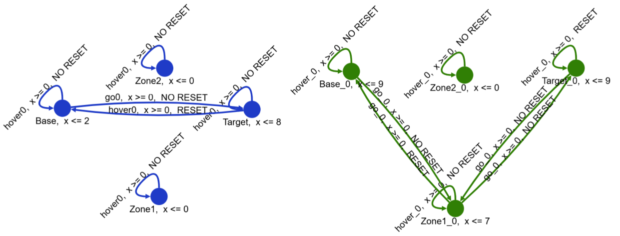

Figure 2: Agents synthesized for $n = z = 2$ (α blue, β green)

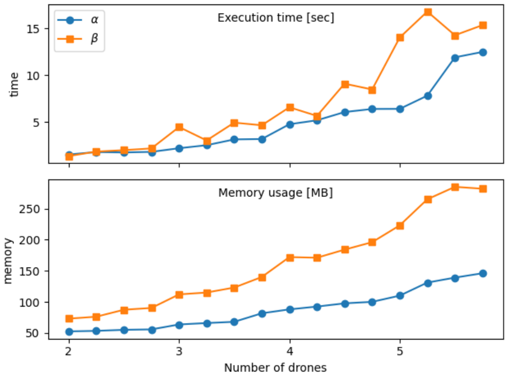

Figure 3: Experimental results

4 FORMAL SYNTHESIS TECHNIQUES

Our tool performs *model synthesis*: given a specification ϕ, it automatically constructs a model M satisfying ϕ. Specifications are expressed in STECTL [4, 5], where the core modality $\langle\langle A\rangle\rangle\gamma$ states that a coalition of agents A has a joint strategy such that all resulting runs satisfy the temporal objective γ. If $\gamma = \mathbf{E}\gamma_1\mathbf{U}_I\gamma_2$, there exists a run where γ_2 holds within interval I, while γ_1 holds beforehand; when $\gamma_1 = true$, we write $\mathbf{EF}\gamma_2$. Similarly, $\mathbf{EG}_I\gamma_1$ requires a run where γ_1 holds throughout I.

The output is a *continuous-time multi-agent system* (CMAS) [4, 5], represented as a *network of timed automata* [1]. The model supports asynchronicity, optional synchronisation, and strong monotonicity, ensuring time progress between action transitions. Agents follow memoryless strategies with imperfect information, selecting actions based only on their current local state [33]. The model is constructed using SMT-based bounded model checking (BMC) [8, 9, 18], encoding the model class, bounded executions, and the specification into an SMT formula [25, 34]. If the encoding is satisfiable, it yields a system model; otherwise, the tool reports that no model exists within the given bounds. The approach supports both full and partial synthesis, including individual agents. *Parametric synthesis* is achieved by introducing parameters as symbolic variables in the SMT encoding. The SMT solver searches for concrete values that satisfy the encoding; parameters may represent model elements or time constraints in the formula, a feature introduced in this class of tools for the first time.

5 ARCHITECTURE AND TECHNOLOGY

The tool consists of the following components (see Fig. 1):

- **Web-based GUI** supports visual editing of formulae and agents, as well as intuitive exploration of synthesis results.
- **BMC Module** encoding synthesis and satisfiability problems into SMT-LIBV2 format, suitable for solving.
- **SMT Solvers** checking satisfiability of generated SMT instances.
- **API Gateway and Microservices** handling identity management, task scheduling, parallel execution, and result storage.

Typical use cases include: **bounded satisfiability** - checking whether a given STECTL formula is satisfiable under given requirements (and yielding a complete model if so); **synthesis** - generating missing parts of the model (e.g., transitions, parameter values, time constraints) that guarantee system correctness; **model checking** and analysis of the synthesised timed multi-agent model.

6 BENCHMARK - UAV MISSION

Our example is based on problems from [14, 23, 32] and considers a surveillance mission involving two cooperating drones (unmanned aerial vehicles) $D1$ and $D2$. The drones must have a strategy ensuring they can reach a target area before deadline d, observe it for a minimum time p, and return safely to base before their fuel f is depleted. These goals are expressed by the following STECTL formula α: $\langle\langle D1, D2\rangle\rangle(\mathbf{EF}_{[0,d)}(tg\wedge\mathbf{EG}_{[0,p)}(tg\wedge\langle\langle D1, D2\rangle\rangle\mathbf{EF}_{[0,f)}base))$, where tg ($base$) denotes that both drones are on the target (at the base), respectively. The target can be reached directly or through one of the fly zones. Formula β: $\langle\langle D1, D2\rangle\rangle(\mathbf{EF}_{[0,\text{inf})}(z1\wedge\mathbf{EF}_{[0,d)}(tg\wedge\mathbf{EG}_{[0,p)}(tg\wedge\langle\langle D1, D2\rangle\rangle\mathbf{EF}_{[0,\text{inf})}(z1\wedge\mathbf{EF}_{[0,f)}base)))))$ states that exists a strategy that both drones reach the target and return via zone1 ($z1$). In our experiment, we define the formula, the locations, hover self-loops, and actions for each agent. Our goal is to synthesize transitions that move drones between locations, time constraints, and parameter values. Fig. 2 presents two agents synthesized from α (blue), with a straight approach to the target and parameter values: $d = 1$, $p = 3$, and $f = 5$, and β (green) with values $d = 1$, $p = 1$, and $f = 5$. Fig. 3 presents the experimental evaluation[1] of our approach for scenarios involving from 2 to 5 agents (n) and fly zones (z). A comparison of the resources required for the synthesis of α and β indicates that the size of the formula has a more pronounced effect on resource consumption than the number of locations and transitions, especially in terms of memory usage. For more details watch our demo video at **https://stectl.ii.uws.edu.pl/demo**.

7 CONCLUSIONS

We presented a tool for synthesising real-time multi-agent systems from STECTL specifications. Its key feature is parametric synthesis, which automatically infers unknown design values, particularly timing parameters, instead of requiring manual early-stage fixation.

---

[1] The experiments were performed on a PC with i7-1065G7 CPU and 16GB RAM

ACKNOWLEDGMENTS

The research by Magdalena Kacprzak was carried out within project no. WZ/WI-IIT/2/2025 at Bialystok University of Technology and financed from the research subsidy of Bialystok University of Technology. Wojciech Penczek was supported by SpaceVote (POLLUX-XI/14/SpaceVote/2023) and PHC Polonium project MoCcA.

REFERENCES

[1] Rajeev Alur and David L. Dill. 1992. The theory of timed automata. In *Real-Time: Theory in Practice*. Springer Berlin Heidelberg, 45–73.

[2] Rajeev Alur, Thomas A. Henzinger, Freddy Y. C. Mang, Shaz Qadeer, Sriram K. Rajamani, and Serdar Tasiran. 1998. MOCHA: Modularity in model checking. In *Computer Aided Verification*, Alan J. Hu and Moshe Y. Vardi (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 521–525.

[3] Étienne André. 2021. IMITATOR 3: Synthesis of timing parameters beyond decidability. In *International Conference on Computer Aided Verification*. Springer, 552–565.

[4] Jaime Arias, Wojciech Jamroga, Wojciech Penczek, Laure Petrucci, and Teofil Sidoruk. 2023. Strategic (Timed) Computation Tree Logic. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023*. ACM, 382–390.

[5] Jaime Arias, Wojciech Jamroga, Wojciech Penczek, Laure Petrucci, and Teofil Sidoruk. 2026. Strategic (timed) computation tree logic. *Autonomous Agents and Multi-Agent Systems* 40, 1 (2026), 7. https://doi.org/10.1007/s10458-025-09726-4

[6] Jaime Arias, Carlos Olarte, Wojciech Penczek, Laure Petrucci, and Teofil Sidoruk. 2024. Model Checking and Synthesis for Strategic Timed CTL using Strategies in Rewriting Logic. In *Proceedings of the 26th International Symposium on Principles and Practice of Declarative Programming* (Milano, Italy) *(PPDP '24)*. Article 10, 14 pages.

[7] Viviana Arrigoni, Giulio Attenni, Novella Bartolini, Matteo Finelli, and Gaia Maselli. 2024. MiKe: Task Scheduling for UAV-based Parcel Delivery. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems* (Auckland, New Zealand) *(AAMAS '24)*. 2138–2140.

[8] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of model checking*. MIT Press.

[9] Armin Biere, Alessandro Cimatti, Edmund M. Clarke, and Yunshan Zhu. 1999. Symbolic Model Checking without BDDs. In *Proceedings of the 5th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS '99)*. Springer-Verlag, Berlin, Heidelberg, 193–207.

[10] Patricia Bouyer. 2009. Model-checking Timed Temporal Logics. *Electronic Notes in Theoretical Computer Science* 231 (2009), 323–341.

[11] Petr Cermák, Alessio Lomuscio, Fabio Mogavero, and Aniello Murano. 2014. MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications. In *Computer Aided Verification - 26th International Conference, CAV 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 18-22, 2014. Proceedings (Lecture Notes in Computer Science, Vol. 8559)*, Armin Biere and Roderick Bloem (Eds.). Springer, 525–532. https://doi.org/10.1007/978-3-319-08867-9_34

[12] Petr Cermák, Alessio Lomuscio, Fabio Mogavero, and Aniello Murano. 2018. Practical verification of multi-agent systems against Slk specifications. *Inf. Comput.* 261 (2018), 588–614. https://doi.org/10.1016/J.IC.2017.09.011

[13] Shushman Choudhury, Kiril Solovey, Mykel Kochenderfer, and Marco Pavone. 2022. Coordinated Multi-Agent Pathfinding for Drones and Trucks over Road Networks. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS '22)*. 272–280.

[14] Jonathan Diller and Qi Han. 2023. Energy-aware UAV Path Planning with Adaptive Speed. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems* (London, United Kingdom) *(AAMAS '23)*. 923–931.

[15] Jonathan Diller, Qi Han, Robert Byers, James Dotterweich, and James Humann. 2025. Hitchhiker's Guide to Patrolling: Path-Finding for Energy-Sharing Drone-UGV Teams. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '25)*. 611–619.

[16] Magdalena Kacprzak, Artur Niewiadomski, and Wojciech Penczek. 2020. SAT-Based ATL Satisfiability Checking. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, Diego Calvanese, Esra Erdem, and Michael Thielscher (Eds.). 539–549. https://doi.org/10.24963/KR.2020/54

[17] Magdalena Kacprzak, Artur Niewiadomski, and Wojciech Penczek. 2021. Satisfiability Checking of Strategy Logic with Simple Goals. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021*. 400–410.

[18] Magdalena Kacprzak, Artur Niewiadomski, Wojciech Penczek, and Andrzej Zbrzezny. 2023. SMT-Based Satisfiability Checking of Strategic Metric Temporal Logic. In *ECAI 2023 - 26th European Conference on Artificial Intelligence (Frontiers in Artificial Intelligence and Applications, Vol. 372)*. IOS Press, 1180–1189.

[19] Mary Koes, Illah Nourbakhsh, and Katia Sycara. 2005. Heterogeneous multirobot coordination with spatial and temporal constraints. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3 (AAAI'05)*. AAAI Press, 1292–1297.

[20] Damian Kurpiewski, Wojciech Jamroga, and Michał Knapik. 2019. STV: Model Checking for Strategies under Imperfect Information. In *Proc. of the Int. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS'19)*. 2372–2374.

[21] Damian Kurpiewski, Mateusz Kamiński, and Wojciech Jamroga. 2024. STV+FLY: On-the-Fly Model Checking of Strategic Ability in Multi-Agent Systems. In *ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024) (Frontiers in Artificial Intelligence and Applications, Vol. 392)*, Ulle Endriss, Francisco S. Melo, Kerstin Bach, Alberto José Bugarín Diz, Jose Maria Alonso-Moral, Senén Barro, and Fredrik Heintz (Eds.). IOS Press, 4483–4486. https://doi.org/10.3233/FAIA241035

[22] Damian Kurpiewski, Witold Pazderski, Wojciech Jamroga, and Yan Kim. 2021. STV+Reductions: Towards Practical Verification of Strategic Ability Using Model Reductions. In *Proc. of the Int. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS'21)*. ACM, 1770–1772.

[23] Mickey Li, Arthur Richards, and Mahesh Sooriyabandara. 2021. Reliability-Aware Multi-UAV Coverage Path Planning using a Genetic Algorithm. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (Virtual Event, United Kingdom) *(AAMAS'21)*. 1584–1586.

[24] Alessio Lomuscio, Hongyang Qu, and Franco Raimondii. 2015. MCMAS: An Open-Source Model Checker for the Verification of Multi-Agent Systems. *International Journal on Software Tools for Technology Transfer* 24 (2015), 84–90. Available online.

[25] Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. 2006. Solving SAT and SAT Modulo Theories: From an abstract Davis–Putnam–Logemann–Loveland procedure to DPLL(T). *J. ACM* 53, 6 (2006), 937–977.

[26] Artur Niewiadomski, Magdalena Kacprzak, Damian Kurpiewski, Michał Knapik, Wojciech Penczek, and Wojciech Jamroga. 2020. MsATL: A Tool for SAT-Based ATL Satisfiability Checking. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '20, Auckland, New Zealand, May 9-13, 2020*. International Foundation for Autonomous Agents and Multiagent Systems, 2111–2113.

[27] Artur Niewiadomski, Maciej Nazarczuk, Mateusz Przychodzki, Magdalena Kacprzak, Wojciech Penczek, and Andrzej Zbrzezny. 2024. SMT4SMTL: A Tool for SMT-Based Satisfiability Checking of SMTL. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2024, Auckland, New Zealand, May 6-10, 2024*, Mehdi Dastani, Jaime Simão Sichman, Natasha Alechina, and Virginia Dignum (Eds.). International Foundation for Autonomous Agents and Multiagent Systems / ACM, 2815–2817. https://doi.org/10.5555/3635637.3663297

[28] Aritra Pal, Anandsingh Chauhan, and Mayank Baranwal. 2025. Together We Rise: Optimizing Real-Time Multi-Robot Task Allocation using Coordinated Heterogeneous Plays. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'25)*. 1586–1594.

[29] Wojciech Penczek, Laure Petrucci, and Teofil Sidoruk. 2025. Satisfiability Checking for (Strategic) Timed CTL Using IMITATOR. In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence, ICAART 2025 - Volume 1, Porto, Portugal, February 23-25, 2025*, Ana Paula Rocha, Luc Steels, and H. Jaap van den Herik (Eds.). SCITEPRESS, 144–155. https://doi.org/10.5220/0013257700003890

[30] Wojciech Penczek and Agata Półrola. 2006. *Advances in Verification of Time Petri Nets and Timed Automata: A Temporal Logic Approach*. Studies in Computational Intelligence, Vol. 20. Springer.

[31] Amir Pnueli. 1977. The Temporal Logic of Programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (SFCS '77)*. IEEE Computer Society, 46–57.

[32] Jürgen Scherer and Bernhard Rinner. 2017. Short and full horizon motion planning for persistent multi-UAV surveillance with energy and communication constraints. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vancouver, BC, Canada). IEEE Press, 230–235.

[33] Pierre Yves Schobbens. 2004. Alternating-time Logic with Imperfect Recall. In *1st International Workshop on Logic and Communication in Multi-Agent Systems (LCMAS 2003) (Electronic Notes in Theoretical Computer Science, Vol. 85)*. Elsevier, 1–12.

[34] Roberto Sebastiani. 2007. Lazy Satisfiability Modulo Theories. *Journal of Satisfiability, Boolean Modeling and Computation* 3, 3-4 (2007), 141–224.

[35] Wiebe van der Hoek and Michael Wooldridge. 2002. Tractable multiagent planning for epistemic goals. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 3* (Bologna, Italy) *(AAMAS '02)*. Association for Computing Machinery, New York, NY, USA, 1167–1174. https://doi.org/10.1145/545056.545095